SSIS Configurations: When and How

[SSIS](http://www.bradleyschacht.com/category/ssis/), [Syndication](http://www.bradleyschacht.com/category/syndication/)

by [Bradley Schacht](http://www.bradleyschacht.com/author/bradleyschacht/)

Configurations in SSIS are a lovely tool that when implemented correctly can greatly reduce package maintenance.  It is very possible to over configure a package, so be sure that you only do what you need to in order to make life easier.  Sometimes too much configuration can equal confusion.  So the big questions, what is a configuration, when should I use it and how do I use it.

Let’s tackle that first one then roll right into the other two.  What is a configuration?  In simplest terms a configuration is a way to modify a property without opening a package.  This can apply to almost everything inside your packages.  In this post we are going to focus out attention on connection managers because that is where configurations are most commonly used.  Think of the properties of a connection manager.  There is a name, a connection string, an initial catalog, server name, username, password and some other things as well. A configuration can be put on one or multiple properties.  So you can configure the server name, which is a very common property to change, or the username and password.  That gets part 1 out of the way (the property part) now for part 2, the modification without opening the package part.  A configuration is going to be stored someplace outside the code that makes up the SSIS package.  This may be in an environment variable on a server, or more commonly in either an XML file on the file system or in a SQL Server table.  When you make a change in the external location the next time the package is run it will use the new value that you entered.

Now that we know what a configuration is (a way to update a package property from outside the package) let’s chat about when to use them.  In general use a configuration whenever you want to be able to change a value stored inside your package without actually having to open the package.  Back to our connection manager example.  The most common place to put a configuration is on the connection string (or just the server name).  The reason for this: as you migrate the package from DEV to QA and ultimately production you have to change the server where the data is being pulled from.  A package running in production using DEV data is absolutely useless.  So, rather than having to open all your packages and change the value of the server name (from ServerDEV to ServerPROD for instance) you can put a configuration on the server name property and change the value inside the configuration file.  Which brings us to part 2 of when you use configuration files: when you have a common value among multiple packages.  50 packages for the same application are going to probably all hit the same set of databases on the same servers.  It is unlikely that all 50 will hit 50 different servers.  So you will have a connection to the AdventureWorks database in all those packages.  Just use the same configuration file to change the server name property on the connection in all your packages and now you only have to change one config file rather than open and change 50 connections. Boom! Time saver!

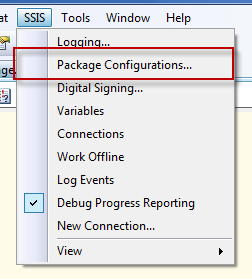
Configs are not limited to only connection managers.  A configuration can be applied to things like the value of a variable, or the executable on an execute process task, all the way down to the sql statement that will be run in an execute SQL task.  We are just using connection managers for this example because they are the most commonly configured.

Finally the reason you are reading this blog, I think…  HOW?!?!?!?!

The how portion is a little more complicated than “when you want to change a value”.  The process for adding a configuration is the same basic process for all 5 types on configs.  The manner in which they are used (ie stored, accessed, maintained, etc) however differs slightly.

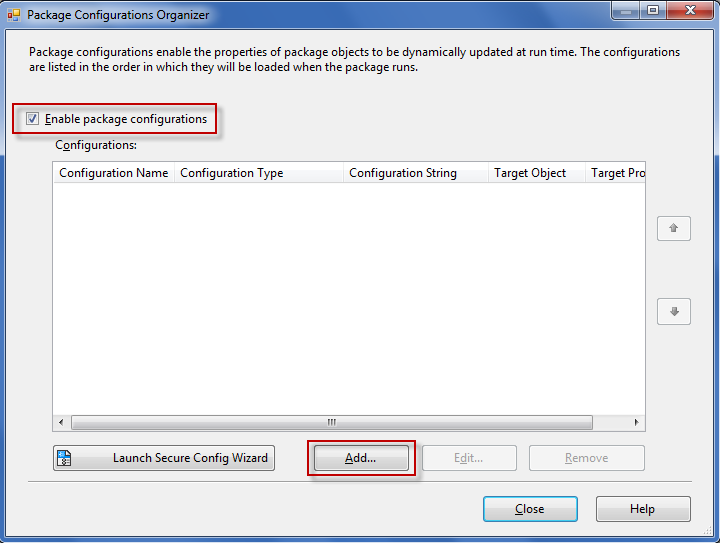
**Step 1: Enable Configurations**

No, configurations are not enabled by default.  What were they thinking?  Why aren’t they enabled?  Why do I have to enable them?  Stop complaining for one moment, it’s only a check box and it is right next to the place where you go to add them anyway.  You’ll already be on the screen, no big deal.  Begin by right-clicking in a blank spot on the Control Flow design pane or click SSIS from the menu at the top of BIDS.  Either way, select Package Configurations from the list of resulting options.

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations1.png)

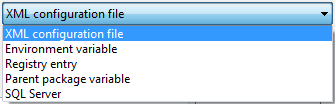
Next enable configurations if they are not already enabled by checking the wildly annoying (just kidding, it’s not annoying and actually can be quite handy) check box in the top left labeled Enable package configurations.

The screen lights up because it is happy to be used and you can click Add at the bottom of the screen to get started.

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations2.png)

**Step 2: Choose Your Own Configuration**

Ok, so a play on words with the old kids books Choose Your Own Adventure didn’t work out as well as I planned in the title of a step, but you get the idea. A fun drop down menu will allow you to choose the type of configuration you want to use.  Let me just quickly give you a rundown of what each one does.  Please note how nicely the list is composed alphabetically so you can quickly find the type you want.  Oh wait, it’s not in alphabetical order?  No. What order is it in? umm….  If you have a good answer for this, please let me know, I have always wondered why they chose the order they did.

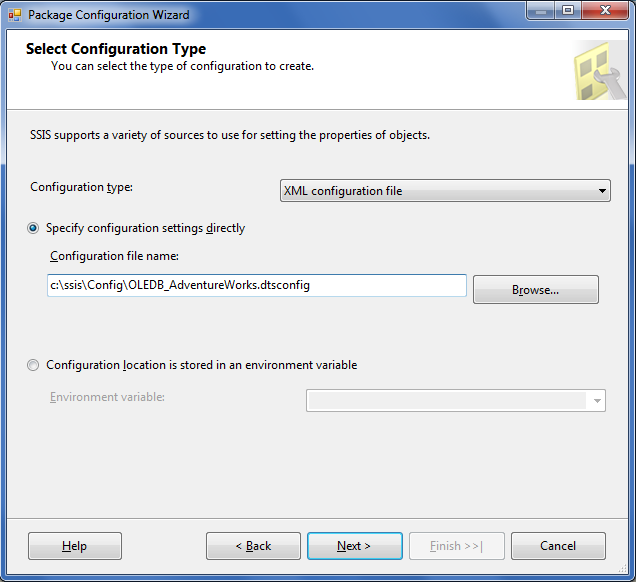
[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations3.png)

* XML – Stores configuration in an XML file (wow, they really chose a great name for that configuration type, Microsoft 1, Horrible Names for Things 0).  This can configure multiple values, but be careful when storing things like passwords because it is all in PLAIN TEXT when you open the file.  There is no native encryption for an XML configuration file.
* Environment Variable – Just like it says, it is an environment variable on the system.  Be sure to check with sys admins before going with this as they may have some problems with you doing this.  Configures only a single property (makes sense, it only stores one value, so it can only config one value)
* Registry entry – See environment variable but replace references to environment variables on the system with in the registry.  Honestly, I have NEVER used this in a real life situation.  No one likes to mess with the registry and they sure don’t like having consultants mess with the registry.
* Parent package variable – Again, a single property configuration.  This is used in parent-child package situations.  Yet again, what a lovely name that matches its use!  This will be chosen in the CHILD package and you supply a variable that is in the PARENT package.  You can take that value, be it a user created or a system variable, and over ride a property in your child package.  Can be pretty useful, I have done this several times, but it is kinda a niche situation.
* SQL Server – I saved the best for last, also it was last on the list.  My personal favorite, it works just like the XML option except the values are stored in a SQL Server table rather than in a file on the file system.  One great advantage is built in backup.  Add your configuration database to your regular backup maintenance plan and you are good to go!

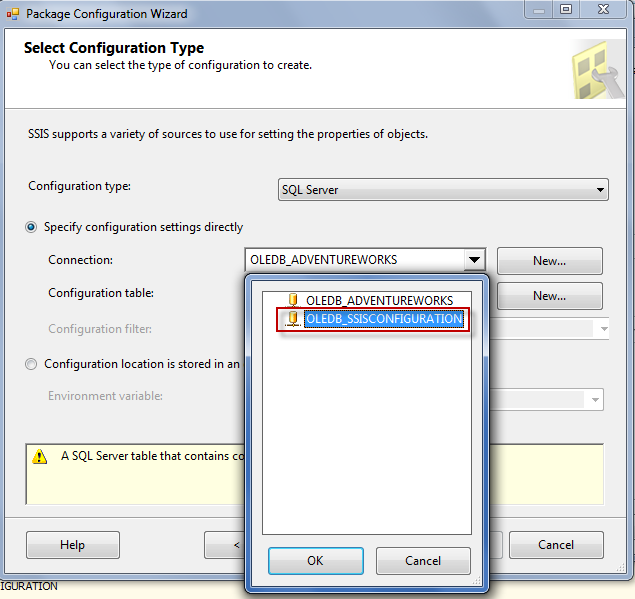
**Step 3: Location or Name**

Once you chose the type of configuration you need to pick out where that configuration will reside (if it is XML or SQL Server) or the name of the source (Name of the registry entry, name of the environment variable, name of the parent variable).

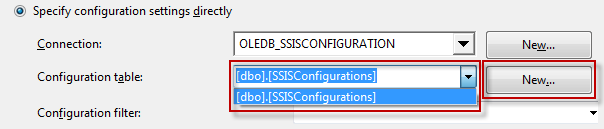
In the case of XML browse to a location on the file system and name your file.  If you are doing this for a connection manager I recommend using the name of the connection manager to be the name of the file that way you can easily identify what it is used for when browsing the file system.  In our example we are configuring the Adventure Works connection that in our package is called OLEDB\_ADVENTUREWORKS, so we will name the file OLEDB\_AdventureWorks.dtsconfig.  If you have a pre-existing configuration browse to it on the file system and select it.

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations4.png)

If going the route of SQL Server you will need a dedicated connection in your package to your configuration database.  I have one called OLEDB\_SSISCONFIGURATION which connects to my SSISConfiguration database.  On the SQL Server settings choose the appropriate connection and then select the table that stores all your configs.

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations5.png)

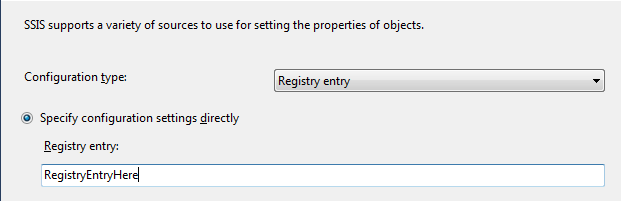
If you don’t already have a configuration table set up you can use the nifty little button that says New next to Configuration table to create one.  It will even automatically script the create table for you!  How nice.  At any rate, select the table that you store the configs in and then continue reading below this image…

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations6.png)

At this point we are just about where you would be with an XML configuration.  Where you normally name a configuration with XML or pick an existing one, you do the same thing with the configuration filter box.  If there are no existing configurations in the table or you want to add a new one type in the box.  If you want an existing one, choose it from the drop down menu.  Either way think of the configuration filter like you think of the name of an XML configuration file.

[http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations7.png](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations7.png)

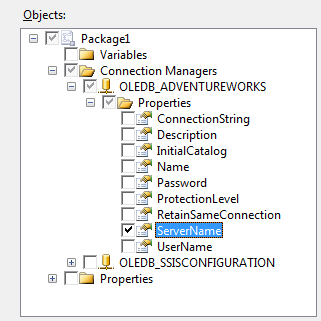
For an environment variable pick the entry from the drop down that hold your value you want to use. For registry entry type in the name of the entry and for parent package variable type the name of the variable in the parent package.

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations8.png)

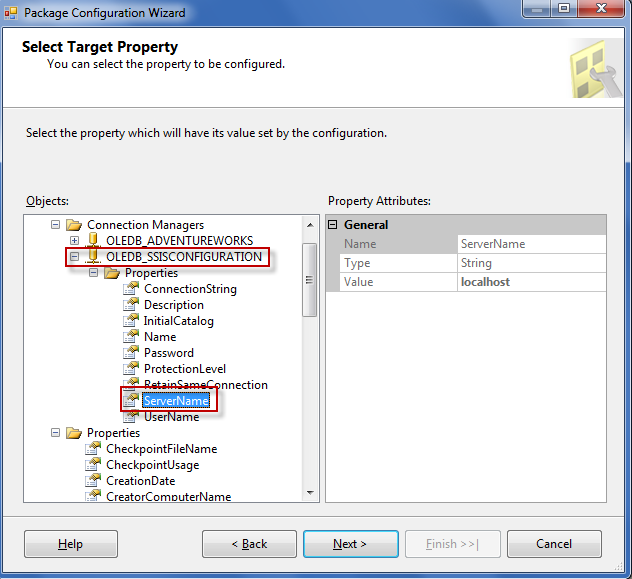
From this point things are the same, the only difference is the number of properties you can configure.

**Step 4: Pick Your Property (Or Properties)**

Our little scenario depicted a world where we were configuring an AdventreWorks connection.  When you click next from the previous screenshots the image below will be what you see.  A list of all the properties in your package that you can configure.  It’s like a configuration buffet.  For XML or SQL Server you can check off as many items as you would like.  Be aware though, that if you check off an item and want to use this configuration in another package, ALL checked properties need to exist.  By that I mean, if you check the option for Server Name on the connection called OLEDB\_ADVENTUREWORKS and the connection OLEDB\_ADVENTUREWORKS doesn’t exist, the package will throw an error.  This is not so much a problem if you go with a single configuration per connection, but if you start to share configuration files with multiple connections or multiple object properties it can become an issue.

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations9.png)

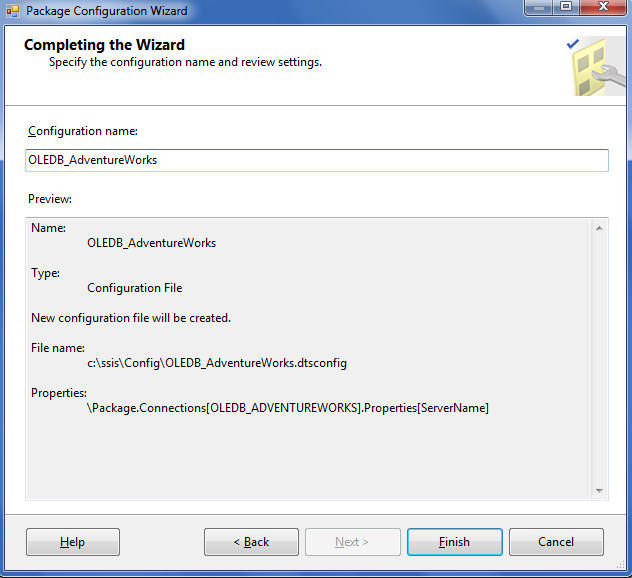
That problem doesn’t arise with Registry, environment variable or parent package variable.  The next image is what you will see there.  Notice that the screen is the same except it is missing the check boxes.  Here you simply highlight the property to configure.  Since these types of configurations can only hold one value, you can only configure one value.

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations10.png)

Make your selections and click Next.

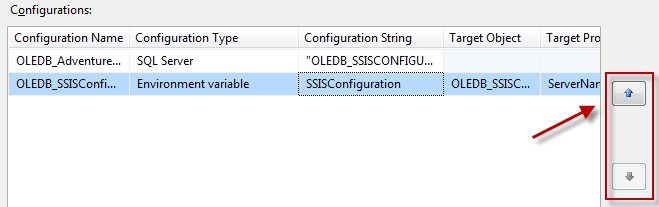
**Step 5: Name that Config**

The final screen will allow you to put a name on your configuration.  This name means nothing at all, that’s right, nothing, except for it helps you know what you put in the configuration.  Again, my recommendation is to make it the same as what you configured, for instance the connection manager name, or Package Variables, or something like that. Now when I look I know that this config entry is for my Adventure Works connection.

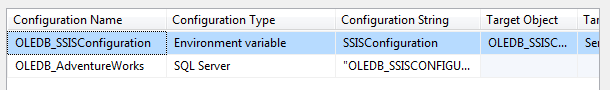
[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations11.png)

One nice feature is that you can add multiple configurations to your package.  Let’s say you do have a SQL Server configuration, there is a connection to that database.  So naturally you need to change the database it looks at.  What do you do?  Add a configuration to tell the connection to your configuration database where to look.  Read that like 3 more times to make sure you understand it before reading further.  Please, this is not a joke, it is a very important concept.  In this case, as you can imagine you will not use a SQL Server configuration.  The most common things to do here are use a single XML file or an environment variable.  I usually set up an environment variable that holds just the connection string to the configuration database.  That environment variable then goes on ALL servers where the package will run and is named the EXACT SAME THING!

Notice in the screenshot below that I used an environment variable called SSISConiguration to configure the ServerName (that gets a little cut off on the far right) or the object OLEDB\_SSISCONFIGURATION (also a little cut off).  This will tell that connection to point to the right environment.  In DEV that connection string will hold the dev server name and credentials.  In QA it will hold the configuration database QA server name and credentials, and same for PROD, but with prod server name and credentials.  I also highlighted the arrows on the right side.  You can move configurations up and down in the list.  This is VERY important because configurations are applied in the order they show in the list.  As it stands our package that we just moved to QA will use go to SQL Server (still pointing to DEV) and configure the value of the Adventure Works connection.  Then it will tell the connection to the configuration database to point to QA.  Those events happened backward and we got bad data.  We need to point to QA configuration database then pull all values out of the configuration database.  Use the up arrow with the proper configuration highlighted to move it to the top of the list.

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations12.png)

Everything should look like this now:

[](http://www.bradleyschacht.com/wp-content/uploads/2011/06/Configurations13.png)

That about wraps up configurations.  Just a couple of wrap up items:

* Configurations are case sensitive: Adding a configuration to OLEDB\_AdventureWorks in one package and then trying to use that same configuration on OLEDB\_ADVENTUREWORKS in another package will not work.  The second package’s values will never get changed.
* If it is in your configuration it needs to be in your package.  This is why I don’t put more than one connection in a configuration.  If you have OLEDB\_AADVENTUREWORKS and OLEDB\_ADVENTUREWORKS\_DW in the same configuration you will have to create a second configuration when you build a package that uses only the regular Adventure Works connection and not the DW.
* My recommendation is to always make all your connection manager names in all caps.  This will eliminate the headache caused by the bullet point above.  No more guessing what another developer did for the case of a connection in their package or having to duplicate configs because of a different case.
* Yes, you will have to edit all your configurations when you move them from environment to environment, but you will only have to do it ONCE and you only have to change ONE configuration per connection, not every single package.
* I like SQL Server for configurations.  Running an UPDATE and REPLACE is much quicker than opening an XML file and updating it.  Even if there are only 10 XML files, the maintenance on SQL Server is much better than the maintenance on XML.
* If you are storing passwords they are in PLAIN TEXT!  They are not added automatically either, if you check the property for password it will add a blank place holder.  Microsoft isn’t responsible for you storing the super secret password of 1234 in plain text for anyone to go get.  I don’t blame them.
* If you are storing passwords they are in PLAIN TEXT!
* One last time, if you are storing passwords they are in PLAIN TEXT!  I am not responsible for you getting the bright idea to store your password in an XML file and then email it to another developer but you acidentially type in someone outside the company’s email address.  Oops, better polish that resume off and I don’t recommend saying that you are an SSIS configuration expert as a bullet point on your last job.

I hope this has helped someone out there with their configuration issues.  If you want specifics on how to set up the individual configuration types please let me know. I’d be more than happy to do a blog with details on how to specifically set type up rather than a summary like this.  It would be detailed, short, sweet and to the point.

I did a webinar on this same topic for Pragmatic Works Free Training on the T’s on June 7, 2011.  If you are interested in seeing that so you can follow along with some demos you can catch it by visiting this blog entry for as long as we keep our webinars on the website.  I don’t know how long we have access to those, but as of this blog it is definitely still working.  Here is the link [SSIS Webinar Recording Is Up](http://www.bradleyschacht.com/ssis-webinar-recording-is-up/).